

Podizanje razvojnog okruženja Izbor omiljenog tekst editor-a.

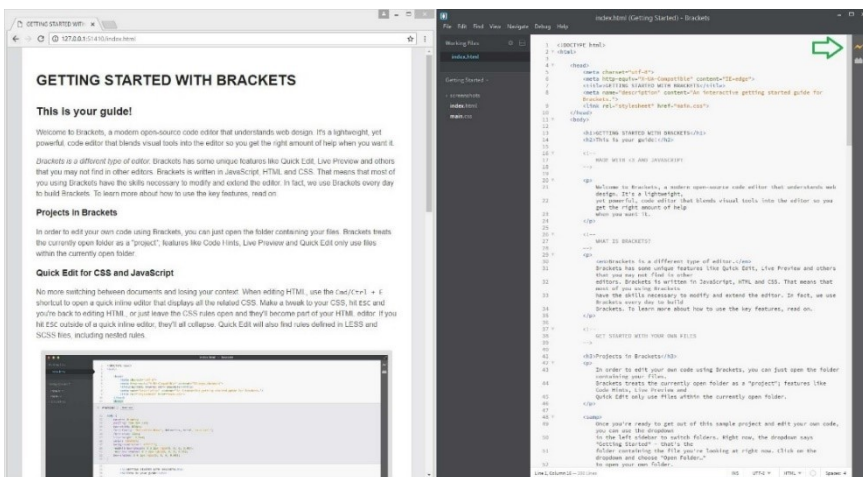
Za rad sa HTML-om i CSS-om vam je ponekad dovoljan i najobičniji tekstualni editor poput Notepad-a. U Notepadu se može upisati kod, File – Save As , tip fajla „All files“, Encoding: „UTF-8“ i naziv fajla PrvaHtmlStranica.html.

```
<!DOCTYPE html>
<html>
<head>
  <title>Pišem HTML iz Notepad-a</title>
</head>
<body>
  <p>Postoji lakši način za pisanje HTML-a.</p>
</body>
</html>
```

Dobijena je prva HTML stranica koja se otvara u vašem omiljenom pretraživaču. Postoji mnogo besplatnih editora koji omogućavaju formatiranje koda u zavisnosti od izabranog programskog jezika, a neki od istaknutih su i Notepad++ i Brackets. VS Code je mnogo napredniji alat i preporuka je da se sve piše u njemu.

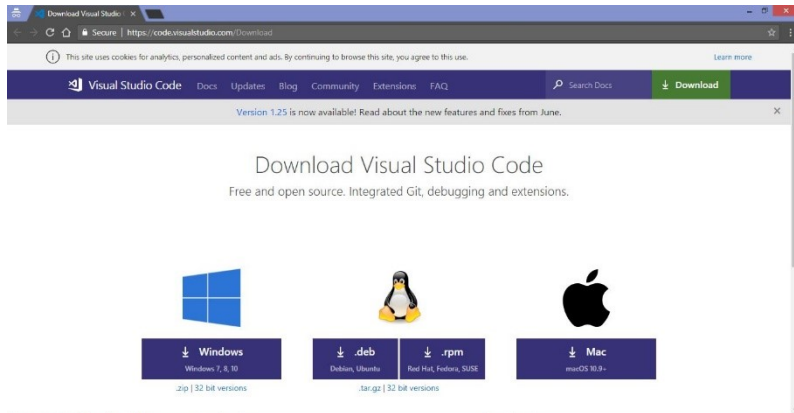
Instalacija Brackets tekst editor-a

Na sajtu <http://brackets.io> je moguće preuzeti moderni i besplatni tekst editor koji olakšava web dizajn. Istaknuta funkcionalnost Brackets-a je opcija „Live Preview“. Navedena opcija omogućava da možete istog trenutka videti kakav uticaj na web stranicu ima to što ste izmenili u CSS, HTML ili nekom drugom fajlu.

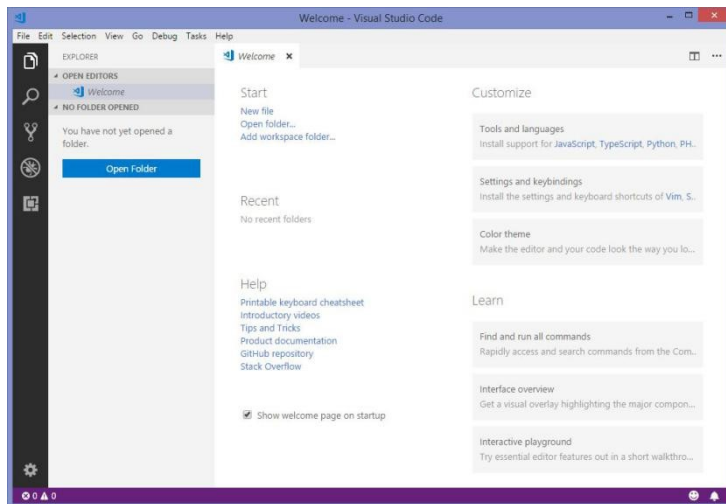


Instalacija Microsoft Visual Studio Code-a

Besplatnu verziju Microsoft-ovog Visual Studio Code-a možete preuzeti sa adrese: <https://code.visualstudio.com/Download> . Izbor verzija za Windows, Linux ili Mac OS, možete videti na slici:

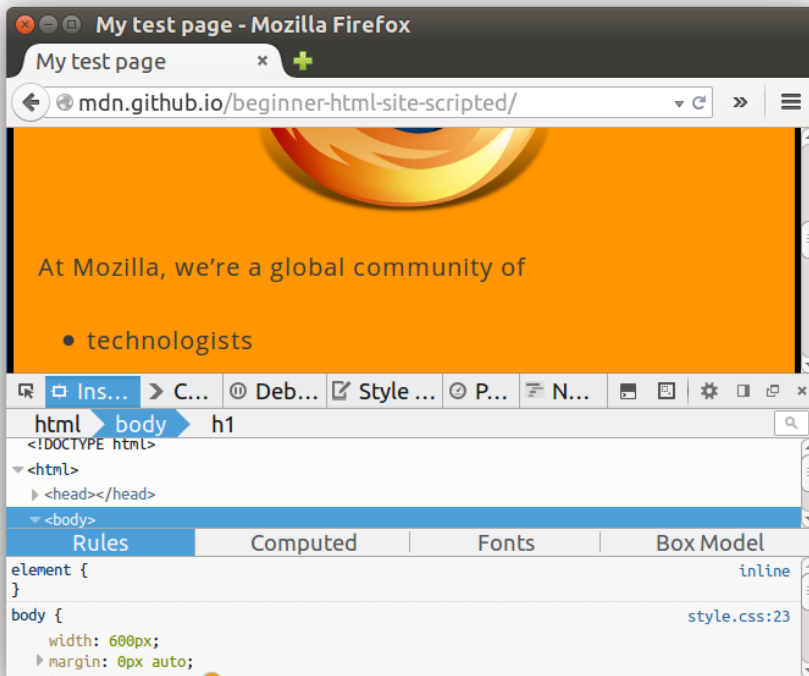


Nakon brze instalacije, pokretanja aplikacije i promene kolor teme, dobija se ekran prikazan na slici:



Programerski alati u browser-u (eng. browser developer tools)

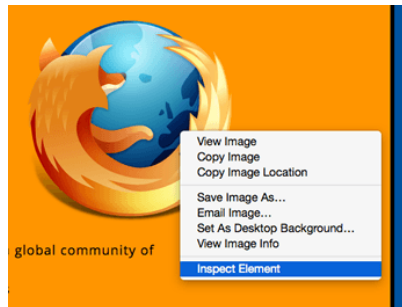
Svaki moderan browser sadrži programerske alate. Služe za inspektovanje trenutno učitano HTML-a, CSS-a i JavaScript-a i pokazuju šta se sve učitalo i koliko je vremena za to trebalo. U daljem tekstu će biti oslovljavani kao „devtools“ i izgledaju kao na slici:



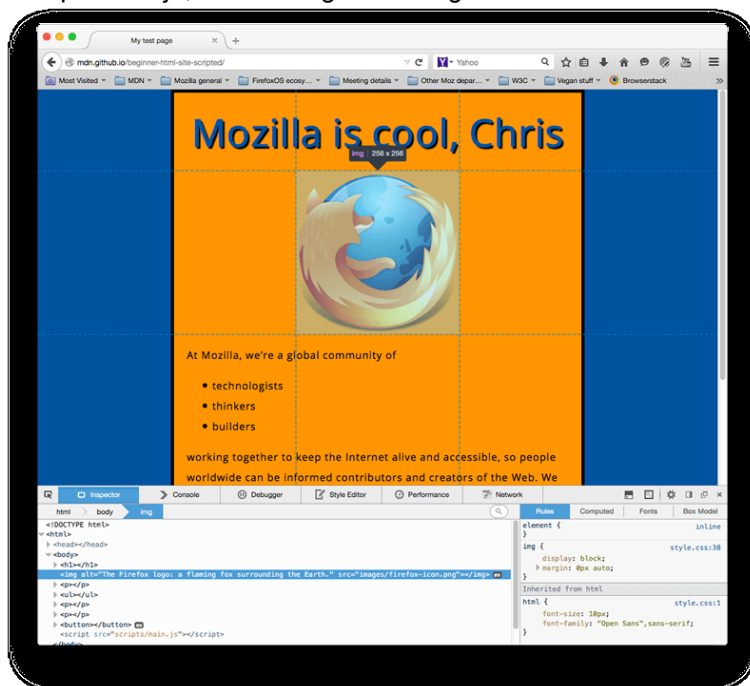
Postoje tri načina za pokretanje:

- **Preko tastature.** „CTRL + Shift + I“, osim
 - **Internet Explorer**-a „F12“,
 - **Mac OS X**-a „⌘ + ⌥ + I“
- **Preko menija.**
 - **Firefox.** Menu, Developer, Toggle tools ili Tools, Web Developer, Toggle tools.
 - **Chrome.** View, Developer, Developer tools.
 - **Safari.** Develop, Show Web Inspector. Ako se Develop meni ne vidi, potrebno je otići u Safari, Preferences, Advanced i čekirati „Show Develop menu in menu bar checkbox“.
 - **Opera.** Developer, Web inspector.

- **Iz kontekst menija.** Desni klik na bilo koji element u web stranici (CTRL + klik na Mac-u) i izbor opcije „Inspect element“ iz kontekst menija koji se pojavi (na ovaj način se odmah označi selektovani element) i prikaže se sledeća slika:



Prilikom pokretanja, devtools uglavnom izgleda kao na slici:



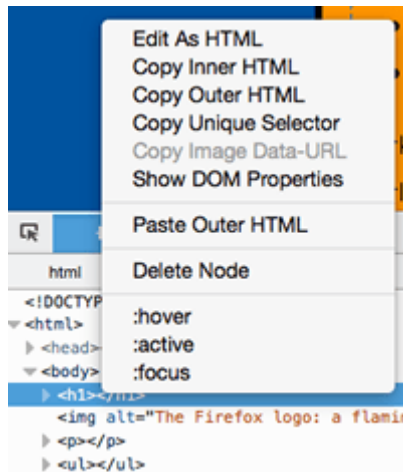
Alat omogućava da se na brz način u browser-u izmeni HTML i CSS i da se prikaže.

Ako ne vidite deo za inspektovanje:

- Pritisnuti/Kliknuti na Inspector tab.
- U Internet Explorer-u, pritisnuti/kliknuti DOM Explorer ili pritisnuti CTRL +1 .

- U Safari-ju možete videti HTML ako niste selektovali nešto drugo da se prikazuje u prozoru. Pritisnuti Style dugme da bi se video CSS.

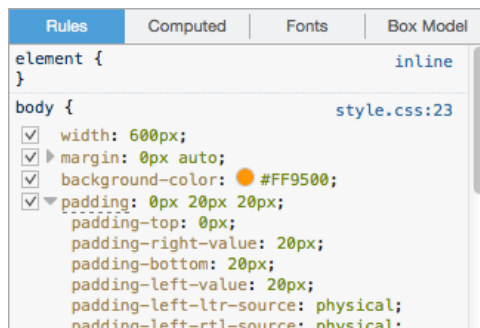
DOM inspector se može pokrenuti desnim klikom (CTRL + klik na Mac-u) na HTML element, nakon čega će se otvoriti kontekst meni. Opcije menija izgledaju drugačije u različitim browser-ima a zajedničke opcije su im uglavnom iste i prikazane su na slici:



Opcije su:

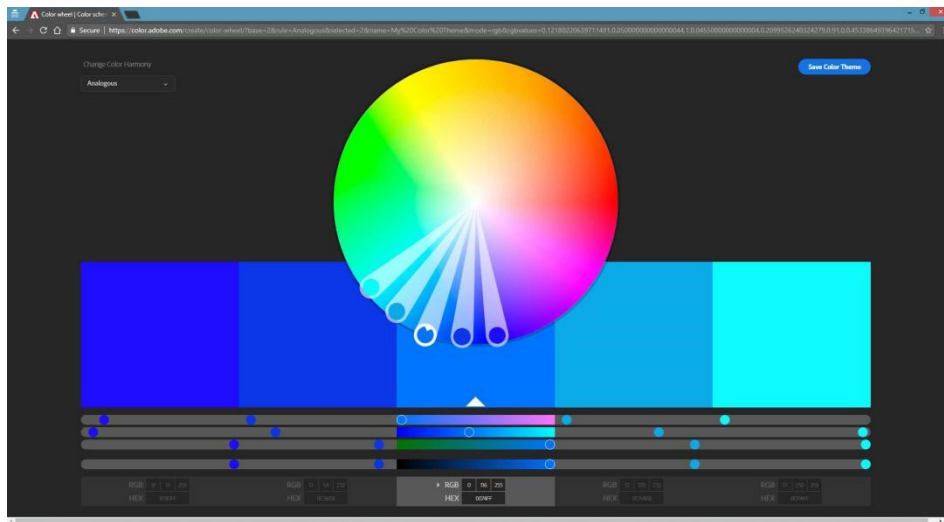
- **Delete Node** (nekad i Delete Element). Briše trenutni element.
- **Edit as HTML** (nekad i Add attributr/Edit text). Omogućava da se promeni HTML i da se u hodu vide rezultati.
- **:hover/:active/:focus** Tera stanja elementa da budu uključena, da bi videli kako će stil izgledati.
- **Copy/Copy as HTML** Kopira trenutno selektovani HTML.

CSS editor prikazuje CSS pravila primenjena na trenutno selektovani element:



Izbor kolor tema pomoću Adobe Color CC online alata

Za vrhunski dizajn su potrebani ili talenat ili godine iskustva i usavršavanja. U ovom praktikumu neće biti reči u dobroj dizajnerskoj praksi. Jedna od bitnijih stvari u dizajnu je izbor boja i ovde će biti ukratko prikazan jedan online alat za izbor istih. Reč je o Adobe Color CC <https://color.adobe.com/create/color-wheel/>



U primeru sa slike smo pomeranjem kruga sa strelicom dobili nekoliko usklađenih nijansi plave boje, izražene HEX vrednostima: 1F0DFF, 0C36E8, 0074FF, 0CAAEE i 0DFAFF.

Validacija HTML-a

Veoma je bitno da HTML kod bude pravilno napisan. Najbolji alat za proveru ove vrste je W3C Validator koji se nalazi na adresi : <https://validator.w3.org> .

HTML 5

Osnove HTML-a

HTML (Hypertext Markup Language) nije programski jezik, već je jezik za označavanje (eng. markup), koji definiše strukturu vašeg sadržaja. HTML se sastoji od niza elemenata koje možete izabrati da zatvorite ili grupišete delove sadržaja. Glavni delovi elementa :

`<p class="komentar-autora">Primer paragrafa</p>`

1. Otvarajuća oznaka: Sastoji se od imena elementa, okruženog sa uglastim zagradama „<p>“
2. Zatvarajuća oznaka: Struktuirana je na sličan način kao i otvarajuća oznaka, jedino što ispred imena elementa sadrži kosu crtu (eng. forward slash) „</p>“.
3. Sadržaj: „Primer paragrafa“.
4. Element: Otvarajuća oznaka, zatvarajuća oznaka i sadržaj zajedno čine element.

Atributi sadrže dodatne informacije o elementu, koji se ne prikazuju sa sadržajem. „class“ je naziv atributa i omogućava da element dobije identifikator, koji će kasnije služiti za pronalazak elementa, a „komentar-autora“ je vrednost atributa.

Ugnežđavanje elemenata (eng. nesting) je situacija kada se element nalazi u elementu. Ukoliko želimo da naglasimo određenu reč u paragrafu, možemo to uraditi na sledeći način:

`<p>Naglašeni primer paragrafa</p>`

Potrebno je obratiti pažnju na način na koji se elementi smeštaju jedan u drugi, ovo je jedan nepravilan primer:

`<p>Nepravilan primer paragrafa</p>`

Prazni elementi nemaju sadržaj, primer je „“:

``

Navedeni prazan element ima dva atributa, ali nema zatvarajuću oznaku „“ i nema unutrašnji sadržaj. Element slike ne obuhvata drugi sadržaj. Njegova uloga je samo da doda sliku u HTML stranu, na određenom mestu.

Individualni HTML elementi nisu korisni. Na primeru ćemo videti kako se elementi kombinuju, da bi dobili HTML stranicu:

```
<!DOCTYPE html>
<html>
<head>
  <title>Naslov</title>
</head>
```

```
<body>
  
</body>
</html>
```

„<!DOCTYPE html>“ označava tip dokumenta.

„<html></html>“ je „html“ element. Poznat je kao početni (eng. root) element i on obuhvata sav sadržaj na stranici.

„<head></head>“ je „head“ element. Služi kao okvir za sav sadržaj koji želite da uključite u HTML stranicu, a koji nije prikazan krajnjem korisniku. To su ključne reči (eng. keywords) i opis (eng. description) koji se pojavljuju kao rezultati pretrage, CSS stilovi, deklaracija seta karaktera i još mnogo toga.

„<body></body>“ je „body“ element. Sastoji se od kompletnog sadržaja koji želite da pokažete korisniku kada posete vašu stranu.

„meta charset="utf-8"“ element podešava set karaktera na UTF-8, koji poseduje karaktere za najveći broj jezika. Može da izađe na kraj sa bilo kakvim tekstualnim sadržajem. Nema razloga da ovo ne podesite i ovo može pomoći da se kasnije ne pojave problemi.

„<title></title>“ je „title“ element. Definiše naslov na stranici koji se prikazuje u tabu pretraživača u kom je strana učitana. Takođe služi kao opis za bookmark.

Elementi naslova (eng. heading) omogućavaju da se definiše šta su naslovi a šta su podnaslovi u sadržaju, kao i kod knjige. HTML ima 6 nivoa naslova <h1>–<h6>, a na primeru su neki:

```
<h1>Naslov stranice</h1>
<h2>Glavni naslov</h2>
<h3>Podnaslov</h3>
```

HTML ima specijalne elemente za liste i uvek se sastoji od bar dva elementa. Najčešće korišćene liste su sortirane (eng. ordered) i nesortirane (eng. unordered):

- U sortiranim listama redosled elemenata je bitan, kao u receptu. Definiše ih „“ element.
- U nesortiranim listama redosled nije bitan, kao spisku za kupovinu. Definiše ih „“ element.

Svaki element unutar liste bilo kog tipa se nalazi unutar „“ elementa:

```
<p>Na FTN-u</p>
<ul>
  <li>inženjeri</li>
  <li>entuzijasti</li>
```



```
<li>kreativni ljudi</li>
</ul>
<p>rade u timu ... </p>
```

Linkovi se formiraju tako što se se elementu „<a>“ doda „<href>“ atribut:

```
<a href="http://www.ftn.uns.ac.rs">Posetite stranicu Fakulteta tehničkih
nauka</a>
```

Kompletna primer je:

```
<!DOCTYPE html>
<html lang="en-US">
  <head >
    <meta charset="utf-8">
    <title>Osnove HTML-a</title>
  </head>
  <body>
    <p>Na FTN-u</p>
    <ul>
      <li>inženjeri</li>
      <li>entuzijasti</li>
      <li>kreativni ljudi</li>
    </ul>
    <p>rade u timu ... </p>
    <a href="http://www.ftn.uns.ac.rs">Posetite stranicu Fakulteta tehničkih
nauka</a>
  </body>
</html>
```

Spisak izdvojenih HTML elemenata

Kompletan spisak HTML elemenata se nalazi na linku u referencama, a navedeni spisak sadrži elemente koji su najčešće korišćeni:

- <link> ,
- <body> ,
- <footer> ,
- <header> ,
- <h1> , <h2> , <h3> , <h4> , <h5> , <h6> ,
- <hgroup> ,
- <blockquote> ,
- <div> ,
- <hr> ,
- ,
- ,
- <p> ,
- ,
- <a> ,
-
 ,
- <cite> ,
- <code> ,
- ,
- <i> ,
- <q> ,
- <s> ,
- <small> ,
- ,
- ,
- ,
- <video> ,
- <noscript> ,
- <script> ,
- ,
- <ins> ,
- <col> ,
- <colgroup> ,
- <table> ,
- <td> ,
- <th> ,
- <tr> ,
- <button> ,
- <form> ,

- `<input>` ,
- `<label>` i
- `<textarea>` .

`<link>` - Element određuje vezu trenutnog dokumenta sa eksternim resursom. Najčešće se koristi za povezivanje sa CSS fajlovima.

`<body>` - Element koji predstavlja sadržaj dokumenta. Može biti samo jedan `<body>` element u dokumentu.

`<footer>` - Element koji predstavlja podnožje najbližeg elementa.

`<header>` - Element koji predstavlja zaglavlje, uglavnom predstavlja uvodni sadržaj.

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` - Elementi predstavljaju šest nivoa naslova.

`<hgroup>` - Element grupiše više `<h1>`–`<h6>` elemenata.

`<blockquote>` - Element označava izdvojeni tekst pod navodnicima.

`<div>` - Element je generički okvir. Nema uticaj na sadržaj ili raspored, sve dok se na njega ne primeni CSS.

`<hr>` - Element predstavlja tematski prekid između elemenata na nivou paragrafa, npr. promena scene u priči.

`` - Element predstavlja stavku u listi. Mora se naći unutar elementa: ``, `` ili `<menu>`.

`` - Element predstavlja sortiranu listu. Tipično je prikazana kao lista sa brojevima.

`<p>` - Element predstavlja paragraf.

`` - Element predstavlja nesortiranu listu. Tipično je prikazana kao lista sa buletima.

`<a>` - Element kreira hiperlink prema: ostalim web stranicama, fajlovima, lokacijama unutar iste strane ili email adresama.

**`
`** - Element služi za prelom teksta. Korisno je kada se piše adresa, gde su novi redovi bitni.

`<cite>` - Element se koristi za opis reference prema citiranom kreativnom radu i mora da sadrži ili naslov ili link prema radu.

`<code>` - Element pokazuje sadržaj na način da se primeti da je to parče kompjuterskog koda.

**** - Element naglašava tekst.

<i> - Element je tipično prikazan u Italic stilu. Dobar je za prikazivanje tehničkih termina i izraza na stranim jezicima.

<q> - Element prikazuje citiran tekst. Većina modernih browser-a okružuje tekst sa navodnicima.

<s> - Element prikazuje precrtani tekst. Služi za prikazivanje sadržaja koji više nije relevantan. Nije prikladan za prikaz izmena u dokumentu, za to se koriste **** i **<ins>** elementi

<small> - Element smanjuje veličinu fonta, npr. sa *large* na *medium*, skroz do browser-ove minimalne veličine. U HTML5 se koristi za komentare i informacije o autorskim pravima.

**** - Element je generički kontejner za formulisani sadržaj, koji se u startu ne prikazuje. Koristi se za grupisanje elemenata (koristeći *class* ili *id* attribute)

**** - Element je tipično prikazan u Bold stilu i označava da sadržaj ima veliku važnost.

**** - Element dodaje sliku u dokument.

<video> - Element dodaje medija plejer, koji podržava puštanje videa u dokumentu.

<noscript> - Element definiše deo HTML-a koji će biti ubačen ukoliko skripta nije podržana ili je opcija „scripting“ isključena u browser-u.

<script> - Element se koristi da doda ili da se referencira izvršni kod.

**** - Element predstavlja deo teksta koji je obrisano iz dokumenta.

<ins> - Element predstavlja deo teksta koji je dodat u dokument.

<col> - Element definiše kolonu u tabeli.

<colgroup> - Element definiše grupu kolona u tabeli.

<table> - Element predstavlja dvodimenzionalnu tabelu koja se sastoji od redova i kolona.

<td> - Element definiše ćeliju u tabeli, koristi se u **<table>** modelu.

<th> - Element definiše ćeliju kao zaglavlje u tabeli.

<tr> - Element definiše red u tabeli. Unutar jednog reda se mogu stavljati **<th>** i **<td>** elementi.

<button> - Element je dugme na koje može da se klikne, koje može biti korišćeno u formama ili negde gde je jednostavno potrebno dugme.

<form> - Element predstavlja deo dokumenta koji sadrži interaktivne kontrole za postavljanje informacija na web server.

<input> - Element se koristi za pravljenje interaktivnih kontrola u formama i služi za prihvatanje podataka koje upisuje korisnik.

<label> - Element predstavlja naslov sa stavku u korisničkom interfejsu.

<textarea> - Element predstavlja kontrolu za višelinijnsko uređivanje teksta, korisnu kada želite da omogućite korisniku da upiše više teksta u slobodnoj formi, npr. forma za komentar.

CSS 3

Osnove CSS-a

CSS (Cascading Style Sheets) je kod koji se koristi za stilizovanje web stranice. Kao i HTML, CSS nije programski jezik. Takođe, nije ni jezik za označavanje (eng. markup). CSS je jezik formatiranja (eng. style sheet). Omogućava da se elementima u HTML dokumentima selektivno dodeljuju stilovi. Selektovanje svih paragrafa na HTML stranici i bojenje teksta u crvenu boju se vrši sa sledećim CSS-om:

```
p {  
  color: red;  
}
```

- Selektor „p“ je HTML element na početku seta pravila. On selektuje elemente koji će biti stilizovani, u ovom slučaju elemente „<p>“. Da bi se stilizovali drugi elementi samo je potrebno promeniti selektor.
- Deklaracija može biti i jedno pravilo kao što je: „color: red;“. Pravilo određuje koje od svojstava elementa želite da stilizujete.
- Nad svojstvima (eng. properties) se primenjuju pravila u CSS-u, u primeru je „color“ svojstvo elementa „<p>“.
- Vrednost svojstva (eng. property value) se piše posle dvotačke, u primeru je „red“.

Navedeni kod možete upisati u tekstualni fajl i sačuvati ga pod nazivom „style.css“, u folderu „styles“.

Sledeći korak je da se napisani CSS primeni na HTML dokument. U fajlu „index.html“ iz primera napisanog u delu: „Osnove HTML-a“, između „<head>“ i „</head>“ oznaka je potrebno dodati sledeći kod:

```
<link href="styles/style.css" rel="stylesheet" type="text/css">
```

Dobija se rezultat prikazan na slici:

Na FTN-u

- inženjeri
- entuzijasti
- kreativni ljudi

rade u timu ...

[Posetite stranicu Fakulteta tehničkih nauka](#)

Prvi CSS je napisan i sada se svi paragrafi prikazuju u crvenoj boji.

Ostali bitni delovi sintakse su:

- Svaki set pravila (osim selektora) se mora nalaziti unutar vitičastih zagrada „{}“.
- U okviru svake deklaracije se mora koristiti dvotačka „:“, da bi se razdvojilo svojstvo od vrednosti.
- U okviru svakog seta pravila mora se koristiti tačka zarez „;“ za razdvajanje deklaracija, npr.

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

Istovremeno se može selektovati više tipova elemenata i postaviti set pravila za sve njih, na sledeći način:

```
p, li, h1 {  
  color: red;  
}
```

Postoje različiti tipovi selektora. Do sada smo gledali samo selektore elemenata.

Ovo su neki od najčešće korišćenih selektora:

Naziv selektora	Šta selektuje	Primer
Selektor elementa (eng. element, tag, type)	Sve HTML elemente zadatog tipa.	p selektuje <p>
Selektor ID-a	Element na strani sa specifičnim ID-jem (na HTML strani jedan ID ide jednom elementu)	#moj-id selektuje <p id="moj-id"> ili
Selektor klase	Jedan ili više elemenata na strani sa specifičnom klasom (može ih biti više na strani)	.moja-klasa selektuje <p class="moja-klasa"> i
Selektor atributa	Jedan ili više elemenata na strani sa specifičnim atributom (može ih biti više na strani)	img[src] selektuje ali ne i
Selektor pseudo- klase	Jedan ili više elemenata na strani, ali samo u specifičnom stanju, npr.	a:hover selektuje <a>, ali samo kada kursor prelazi preko elementa

	prilikom prelaska kursora preko elementa	
--	---	--

Postoji opcija za promenu fonta, a vrši se dodavanjem sledećeg koda u „index.html“:

```
<link href='https://fonts.googleapis.com/css?family=Open+Sans'
rel='stylesheet' type='text/css'>
```

Sledeći korak je da se sadržaj fajla „style.css“ isprazni, a zatim da se doda kod:

```
html {
    font-size: 10px; /*veličina fonta je 10 piksela */
    font-family: 'Open Sans', sans-serif;
}
```

U CSS dokumentu, sve što se nalazi između /* i */ predstavlja komentar, koji pretraživač ignoriše prilikom interpretacije koda. Ovo je pravo mesto za pisanje beleški.

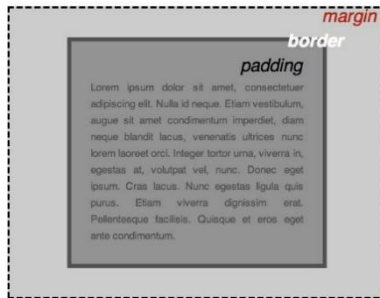
Sledeći primer obuhvata elemente „<h1>“, „“ i „<p>“ u kom centriramo tekst naslova, podešavamo visinu linije i razmak između slova, da bi tekst bio čitljiviji.

```
h1 {
    font-size: 60px;
    text-align: center;
}

p, li {
    font-size: 16px;
    line-height: 2;
    letter-spacing: 1px;
}
```

Jedna stvar koju ćete primetiti kod CSS-a je da je sve u okvirima (eng. boxes), podešavanje njihove veličine, boje, pozicije itd. Većina elemenata na stranici može da se zamisi kao mnogo okvira koji se nalaze jedni na drugima. CSS raspored (eng. layout) se zasniva na tzv. box model-u. Svaki od blokova koji zauzima mesto na strani, ima sledeća svojstva i prikazan je na slici:

- padding, prostor oko sadržaja (npr. oko teksta),
- border, linija koja se nalazi oko padding-a i
- margin, prostor oko (izvan) elementa.



U nastavku će biti opisana sledeća svojstva:

- width (elementa),
- background-color, boja iza sadržaja elementa i padding-a,
- color, boja sadržaja elementa (najčešće teksta),
- text-shadow (postavlja senku na tekstu unutar elementa) i
- display (podešava režim prikaza elementa).

Boja pozadine za celu stranu se menja sledećim kodom:

```
html {
  background-color: #00539F;
}
```

Podešavanje „<body>“ elementa se vrši sledećim kodom:

```
body {
  width: 600px;
  margin: 0 auto;
  background-color: #FBD5A0;
  padding: 0 20px 20px 20px;
  border: 5px solid black;
}
```

- width: 600px – postavlja širinu na 600 piksela,
- margin: 0 auto – prilikom postavljanja dve vrednosti nad svojstvima kao što su margin i padding, prva vrednost se tiče gornje i donje strane elementa (0 u ovom slučaju) a druga vrednost leve i desne strane (ovde je „auto“ specijalna vrednost koja deli dostupan horizontalni prostor podjednako sa obe strane). Takođe se može koristiti jedna vrednost, kao i tri i četiri.

- **background-color:** #FBD5A0- podešava boju pozadine, u ovom slučaju elementa „<body>“ u narandžastu, za razliku od plave koja je podešena za „<html>“ element.
- **padding:** 0 20px 20px 20px – u ovom slučaju se koriste četiri vrednosti za pravljenje prostora oko sadržaja, na vrhu bez padding-a i po 20 piksela sa ostalih strana. Vrednosti se postavljaju na vrhu, desno, dole i levo, u navedenom redosledu.
- **border:** 5px solid black – postavlja pun okvir debljine 5 piksela na svim stranama elementa „<body>“.

Pozicioniranje i stilizovanje naslova sa glavne strane se vrši sledećim kodom:

```
h1 {
    margin: 0;
    padding: 20px 0;
    color: #00539F;
    text-shadow: 3px 3px 1px black;
}
```

Podrazumevana vrednost za marginu naslova je izuzetno velika, zbog čega se javlja velika praznina na vrhu stranice. Ukoliko ne postavite nikakav CSS, nju nameću browser-i. Podrazumevano stilizovanje se poništi kada postavimo vrednost, u ovom slučaju „margin: 0“. Zatim se postavlja gornji i donji padding naslova od 20 piksela i tekst naslov dobija istu boju kao i pozadina „<html>“ elementa.

Jedno vrlo interesantno svojstvo je „text-shadow“, koji daje senku elementu. Četiri vrednosti iz primera su:

- prva vrednost izražena u pikselima podešava „horizontal offset“ senke od teksta, a negativna vrednost je pomera ulevo.
- druga vrednost izražena u pikselima podešava „vertical offset“ senke od teksta, a negativna vrednost je pomera na gore.
- treća vrednost izražena u pikselima podešava „blur radius“ senke – veća vrednost će dati mutniju senku.
- četvrta vrednost podešava osnovnu boju senke.

Centriranje slike se vrši pomoću sledećeg koda:

```
img {
    display: block;
    margin: 0 auto;
```

```
}
```

„<body>“ element je na nivou bloka, što znači da zauzima određeni okvir i može imati margine i ostale vrste razmaka. Element „“ nema navedene mogućnosti zato što pripada grupi „inline“ elemenata. Iz navedenog razloga se elementu „“ upotrebom koda „display: block;“ omogućava da se podešava na nivou bloka.

Sadržaj novog „index.html“ fajla je sledeći:

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8">
    <title>Osnove CSS-a</title>
    <link href="http://fonts.googleapis.com/css?family=Open+Sans'
rel='stylesheet' type='text/css'">
    <link href="styles/style.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    <h1>Fakultet tehničkih nauka</h1>

    
    <p>Na FTN-u</p>
    <ul>
      <li>inženjeri</li>
      <li>entuzijasti</li>
      <li>kreativni ljudi</li>
    </ul>
    <p>rade u timu ... </p>
    <a href="http://www.ftn.uns.ac.rs">Posetite stranicu Fakulteta tehničkih
nauka</a>
  </body>
</html>
```

Sadržaj novog „style.css“ fajla je sledeći:

```
html {
  font-size: 10px;
```

```
font-family: 'Open Sans', sans-serif;  
}
```

```
p, li {  
font-size: 16px;  
line-height: 2;  
letter-spacing: 1px;  
}
```

```
html {  
background-color: #00539F;  
}
```

```
body {  
width: 600px;  
margin: 0 auto;  
background-color: #FBD5A0;  
padding: 0 20px 20px 20px;  
border: 5px solid black;  
}
```

```
h1 {  
margin: 0;  
padding: 20px 0;  
color: #00539F;  
text-shadow: 3px 3px 1px black;  
font-size: 40px;  
text-align: center;  
}
```

```
img {  
display: block;  
margin: 0 auto;  
}
```

Nakon raspoređivanja koda u odgovarajuće fajlove i foldere, dobija se sadržaj strane prikazan na slici.



Spisak izdvojenih CSS svojstava

Kompletan spisak CSS elemenata se nalazi na linku u referencama, a navedeni spisak sadrži elemente koji su najčešće korišćeni, to su:

- background,
- border,
- clear,
- height,
- letter-spacing,
- line-height,
- list-style,
- margin,
- overflow,
- padding,
- position,
- text-align,
- text-indent,
- vertical-align,
- visibility,
- width i
- z-index

background – Svojstvo je skraćen način za deklaraciju svih opcija za pozadinu istovremeno, uključujući: boju, sliku, izvor i veličinu, metod ponavljanja i ostale. Istovremeno može da definiše sledeća svojstva: **background-clip**, **background-color**, **background-image**, **background-origin**, **background-position**, **background-repeat**, **background-size** i **background-attachment**.

Primer HTML koda:

```
<p class="topbanner">  
  Starry sky  
  <br/> Twinkle twinkle
```

```

    <br/> Starry sky
</p>
<p class="warning">Here is a paragraph
<p>

```

Primer CSS koda:

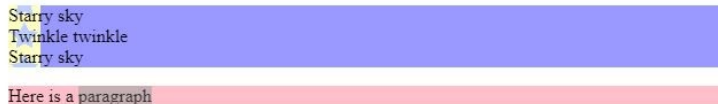
```

.warning {
    background: pink;
}

.topbanner {
    background:
url("https://mdn.mozillademos.org/files/11983/starsolid
.gif") #99f repeat-y fixed;
}

```

Rezultat je prikazan na slici:



border – Svojstvo je skraćen način za podešavanje individualnih vrednosti za border. Istovremeno može da se podese sledeća svojstva: **border-width**, **border-style**, i **border-color**.

Primer HTML koda:

```

<div>I have a border, an outline, AND a box shadow!
Amazing, isn't it?</div>

```

Primer CSS koda:

```

div {
    border: 0.5rem outset pink;
}

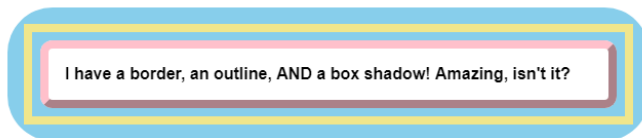
```

```

outline: 0.5rem solid khaki;
box-shadow: 0 0 0 2rem skyblue;
border-radius: 12px;
font: bold 1rem sans-serif;
margin: 2rem;
padding: 1rem;
outline-offset: 0.5rem;
}

```

Rezultat je prikazan na slici:



clear – Svojstvo određuje da li se element može nalaziti pored elemenata koji se nalaze pre njega ili se mora pomeriti dole tj. krenuti od čistog (eng. cleared). Može da sadrži sledeće vrednosti: none, left, right, both, inline-start i inline-end.

Primer HTML koda:

```

<div class="wrapper">
  <p class="black">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit. Phasellus sit amet diam.
Duis mattis varius dui. Suspendisse eget dolor. Fusce
pulvinar lacus ac dui.</p>
  <p class="red">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit. Phasellus sit amet diam.
Duis mattis varius dui. Suspendisse eget dolor.</p>
  <p class="both">This paragraph clears both.</p>
</div>

```

Primer CSS koda:

```

.wrapper{

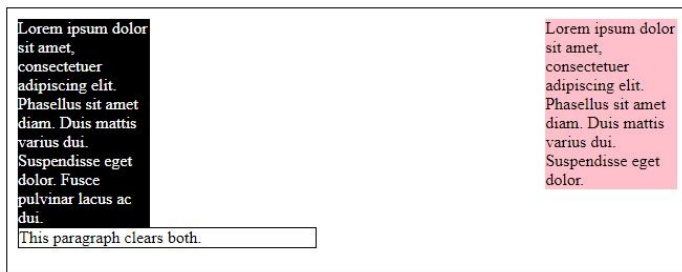
```

```

border:1px solid black;
padding:10px;
}
.both {
border: 1px solid black;
clear: both;
}
.black {
float: left;
margin: 0;
background-color: black;
color: #fff;
width:20%;
}
.red {
float: right;
margin: 0;
background-color: pink;
width:20%;
}
p {
width: 45%;
}

```

Rezultat je prikazan na slici:



height – Svojstvo određuje visinu elementa. Svojstva „min-height“ i „max-height“ imaju prednost nad „height“.

Primer HTML koda:


```
<div id="taller">I'm 50 pixels tall.</div>
<div id="shorter">I'm 25 pixels tall.</div>
<div id="parent">
  <div id="child">
    I'm half the height of my parent.
  </div>
</div>
```

Primer CSS koda:

```
div {
  width: 250px;
  margin-bottom: 5px;
  border: 2px solid blue;
}

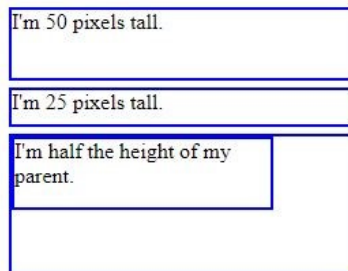
#taller {
  height: 50px;
}

#shorter {
  height: 25px;
}

#parent {
  height: 100px;
}

#child {
  height: 50%;
  width: 75%;
}
```

Rezultat je prikazan na slici:



letter-spacing – Svojstvo određuje razmak između karaktera.

Primer HTML koda:

```
<p class="normal">letter spacing</p>
<p class="em-wide">letter spacing</p>
<p class="em-wider">letter spacing</p>
<p class="em-tight">letter spacing</p>
<p class="px-wide">letter spacing</p>
```

Primer CSS koda:

```
.normal { letter-spacing: normal; }
.em-wide { letter-spacing: 0.4em; }
.em-wider { letter-spacing: 1em; }
.em-tight { letter-spacing: -0.05em; }
.px-wide { letter-spacing: 6px; }
```

Rezultat je prikazan na slici:

```
letter spacing
letter spacing
letter spacing
letter spacing
letter spacing
```

line-height – Svojstvo određuje količinu prostora koji se koristi npr. za visinu linije u kojoj se nalazi tekst.

Primer HTML koda:

```
<div class="box green">
  <h1>Avoid unexpected results by using unitless line-
height.</h1>
  length and percentage line-heights have poor
inheritance behavior ...
</div>
```

```
<div class="box red">
  <h1>Avoid unexpected results by using unitless line-
height.</h1>
  length and percentage line-heights have poor
inheritance behavior ...
</div>
```

```
<!-- The first <h1> line-height is calculated
from its own font-size (30px × 1.1) = 33px -->
<!-- The second <h1> line-height results from
the red div's font-size (15px × 1.1) = 16.5px,
probably not what you want -->
```

Primer CSS koda:

```
.green {
  line-height: 1.1;
  border: solid limegreen;
}

.red {
  line-height: 1.1em;
  border: solid red;
}

h1 {
  font-size: 30px;
}
```

```
.box {
  width: 18em;
  display: inline-block;
  vertical-align: top;
  font-size: 15px;
}
```

Rezultat je prikazan na slici:



list-style – Svojstvo je skraćen način za postavljanje svih vrednosti koje definišu način prikaza liste: **list-style-type**, **list-style-image** i **list-style-position**

Primer HTML koda:

```
List 1
<ul class="one">
  <li>List Item1</li>
  <li>List Item2</li>
  <li>List Item3</li>
</ul>
List 2
<ul class="two">
  <li>List Item A</li>
  <li>List Item B</li>
  <li>List Item C</li>
</ul>
```

Primer CSS koda:

```
.one {
  list-style: circle;
}

.two {
  list-style: square inside;
}
```

Rezultat je prikazan na slici:

- List 1
- List Item1
 - List Item2
 - List Item3
- List 2
- List Item A
 - List Item B
 - List Item C

margin – Svojstvo određuje vrednost oblasti margine na sve četiri strane elementa. Može istovremeno da postavi sledeće margine: **margin-top**, **margin-right**, **margin-bottom** i **margin-left**. Svojstvo može biti definisano sa: jednom, dve, tri ili četiri vrednosti. Primeri:

- Ukoliko je zadata **jedna** vrednost, primenjuje se **na sve četiri strane**.
- Ukoliko su zadate **dve** vrednosti, prva se primenjuje na **gornju i donju**, a druga na **levu i desnu stranu**.
- Ukoliko su zadate **tri** vrednosti, prva se primenjuje na **gornju**, druga na **levu i desnu**, a treća na **donju stranu**.
- Ukoliko su zadate **sve četiri** vrednosti, primenjuje se na gornju, desnu, donju i levu, kao kazaljka na satu (eng. clockwise).

Primer HTML koda:

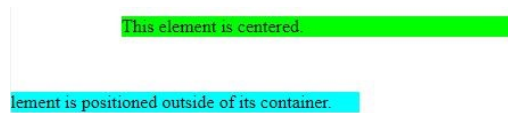
```
<div class="center">This element is centered.</div>

<div class="outside">This element is positioned outside
of its container.</div>
```

Primer CSS koda:

```
.center {  
  margin: auto;  
  background: lime;  
  width: 66%;  
}  
  
.outside {  
  margin: 3rem 0 0 -3rem;  
  background: cyan;  
  width: 66%;  
}
```

Rezultat je prikazan na slici:



overflow – Svojstvo određuje šta da se radi ukoliko je sadržaj elementa prevelik za element u kom se nalazi. Predstavlja skraćen način za postavljanje **overflow-x** i **overflow-y** svojstava. Može se definisati sa jednom ili dve ključne reči: **visible**, **hidden**, **scroll**, **auto** ili **overlay**. Ako su navedene dve ključne reči, prva se odnosi na **overflow-x**, a druga na **overflow-y**. U suprotnom, **overflow-x** i **overflow-y** se podešavaju na istu vrednost.

Primer HTML koda:

```
<p>Michaelmas term lately over, and the Lord Chancellor  
sitting in Lincoln's Inn Hall. Implacable November  
weather. As much mud in the streets as if the waters  
had but newly retired from the face of the earth.</p>
```

Primer CSS koda:

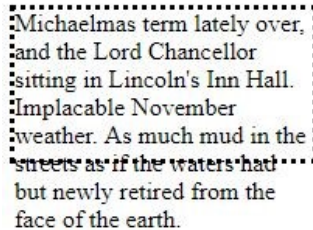
```
p {  
  width: 12em;  
}
```

```

height: 6em;
border: dotted;
overflow: visible; /* content is not clipped */
}

```

Rezultat je prikazan na slici:



Michaelmas term lately over,
and the Lord Chancellor
sitting in Lincoln's Inn Hall.
Implacable November
weather. As much mud in the
streets as if the waters had
but newly retired from the
face of the earth.

padding – Svojstvo određuje prostor oko sadržaja sa sve četiri strane oko elementa. Može istovremeno da postavi sledeće prostore oko sadržaja: **padding-top**, **padding-right**, **padding-bottom** i **padding-left**. Svojstvo može biti definisano sa: jednom, dve, tri ili četiri vrednosti.

Primer HTML koda:

```

h4 {
  background-color: lime;
  padding: 20px 50px;
}

h3 {
  background-color: cyan;
  padding: 110px 50px 50px 110px;
}

```

Primer CSS koda:

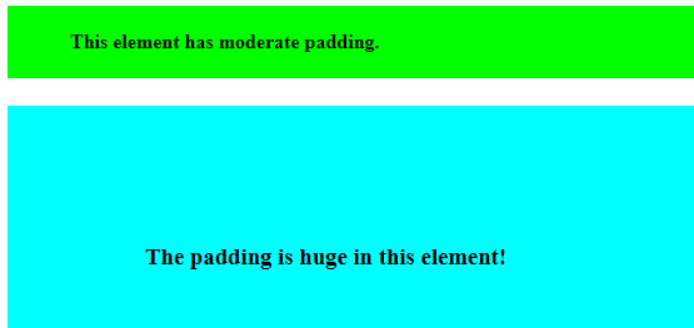
```

<h4>This element has moderate padding.</h4>

<h3>The padding is huge in this element!</h3>

```

Rezultat je prikazan na slici:



position – Svojstvo određuje način na koji je element pozicioniran u dokumentu. Svojstva: **top**, **right**, **bottom** i **left** određuju finalnu lokaciju pozicioniranih elemenata. Tipovi pozicioniranja:

- Pozicionirani element može imati vrednost: **relative**, **absolute**, **fixed** ili **sticky**.
- Relativno pozicionirani element ima izračunatu poziciju i vrednost **relative**.
- Apsolutno pozicionirani element ima izračunatu poziciju i vrednost ili **absolute** ili **fixed**. Svojstva: **top**, **right**, **bottom** i **left** definišu ofset od stranica bloka u kom se element nalazi. Ako element ima margine, one se dodaju u ofset.
- Zalepljeno (eng. sticky) pozicionirani element ima izračunatu poziciju i vrednost **sticky**. Tretira se kao relativno pozicionirani sve dok blok u kom se nalazi ne pređe određeni prag.

Primer HTML koda:

```
<dl>
  <div>
    <dt>A</dt>
    <dd>Andrew W.K.</dd>
    <dd>Apparat</dd>
    <dd>Arcade Fire</dd>
    <dd>At The Drive-In</dd>
    <dd>Aziz Ansari</dd>
  </div>
```



```

<div>
  <dt>C</dt>
  <dd>Chromeo</dd>
  <dd>Common</dd>
  <dd>Converge</dd>
  <dd>Crystal Castles</dd>
  <dd>Cursive</dd>
</div>
<div>
  <dt>E</dt>
  <dd>Explosions In The Sky</dd>
</div>
<div>
  <dt>T</dt>
  <dd>Ted Leo & The Pharmacists</dd>
  <dd>T-Pain</dd>
  <dd>Thrice</dd>
  <dd>TV On The Radio</dd>
  <dd>Two Gallants</dd>
</div>
</dl>

```

Primer CSS koda:

```

* {
  box-sizing: border-box;
}

dl > div {
  background: #FFF;
  padding: 24px 0 0 0;
}

dt {
  background: #B8C1C8;
  border-bottom: 1px solid #989EA4;
  border-top: 1px solid #717D85;
  color: #FFF;
}

```

```

font: bold 18px/21px Helvetica, Arial, sans-serif;
margin: 0;
padding: 2px 0 0 12px;
position: -webkit-sticky;
position: sticky;
top: -1px;
}

dd {
font: bold 20px/45px Helvetica, Arial, sans-serif;
margin: 0;
padding: 0 0 0 12px;
white-space: nowrap;
}

dd + dd {
border-top: 1px solid #CCC;
}

```

Rezultat je prikazan na slici:

C
Crystal Castles
Cursive
E
Explosions In The Sky
T
Ted Leo & The Pharmacists
T-Pain

text-align – Svojstvo određuje kako će linijski sadržaj kao što je tekst, biti poravnat unutar elementa u kom se nalazi. On ne kontroliše poravnanje elemenata, već njihovog sadržaja. Definiše se jednom od ključnih reči: **start**, **end**, **left**, **right**, **center**, **justify** i **match-parent**.

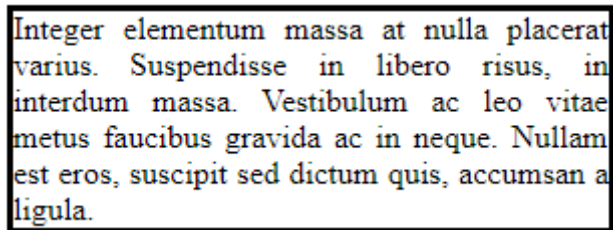
Primer HTML koda:

```
<p class="example">
    Integer elementum massa at nulla placerat
varius.
    Suspendisse in libero risus, in interdum massa.
    Vestibulum ac leo vitae metus faucibus gravida
ac in neque.
    Nullam est eros, suscipit sed dictum quis,
accumsan a ligula.
</p>
```

Primer CSS koda:

```
.example {
    text-align: justify;
    border: solid;
}
```

Rezultat je prikazan na slici:



text-indent – Svojstvo određuje količinu praznog prostora koji se postavlja pre teksta. Podrazumevano je da se prazan prostor postavi samo u prvi red, ali se to može promeniti upotrebom ključnih reči **hanging** i **each-line**.

Primer HTML koda:

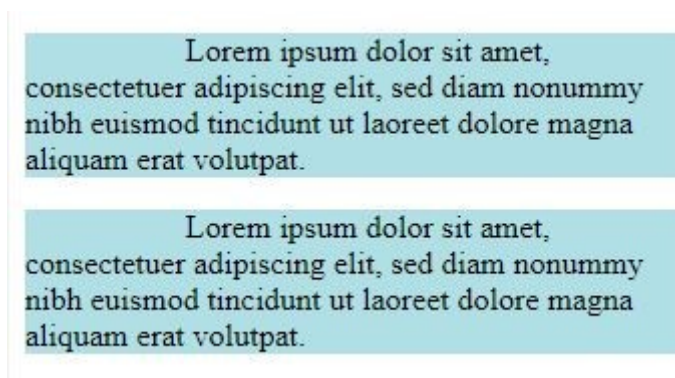
```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed diam nonummy
```

```
nibh euismod tincidunt ut laoreet dolore magna  
aliquam erat volutpat.</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing  
elit, sed diam nonummy  
nibh euismod tincidunt ut laoreet dolore magna  
aliquam erat volutpat.</p>
```

Primer CSS koda:

```
p {  
  text-indent: 5em;  
  background: powderblue;  
}
```

Rezultat je prikazan na slici:



vertical-align – Svojstvo određuje vertikalno poravnanje unutar okvira.

Primer HTML koda:

```
<div>An  image with  
a default alignment.</div>  
  
<div>An  image with  
a text-top alignment.</div>
```

```

<div>An  image with
a text-bottom alignment.</div>
<div>An  image with
a middle alignment.</div>

```

Primer CSS koda:

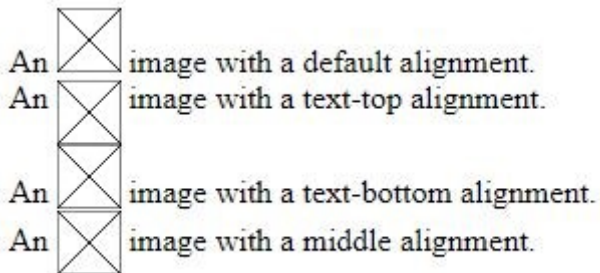
```

img.top { vertical-align: text-top; }

img.bottom { vertical-align: text-bottom; }
img.middle { vertical-align: middle; }

```

Rezultat je prikazan na slici:



visibility – Svojstvo može da pokaže ili sakrije element bez uticaja na raspored u dokumentu. Ovo svojstvo takođe može da sakrije redove ili kolone u elementu „<table>“.

Primer HTML koda:

```

<p class="visible">The first paragraph is visible.</p>

<p class="not-visible">The second paragraph is NOT
visible.</p>

```

`<p class="visible">The third paragraph is visible. Notice the second paragraph is still occupying space.</p>`

Primer CSS koda:

```
.visible {  
    visibility: visible;  
}  
  
.not-visible {  
    visibility: hidden;  
}
```

Rezultat je prikazan na slici:

The first paragraph is visible.

The third paragraph is visible. Notice the second paragraph is still occupying space.

width – Svojstvo određuje širinu elementa. Svojstva „min-width“ i „max-width“ imaju prednost nad „width“.

Primer HTML koda:

```
<div class="px_length">Width measured in px</div>  
<div class="em_length">Width measured in em</div>
```

Primer CSS koda:

```
.px_length {  
    width: 200px;  
    background-color: red;  
    color: white;  
    border: 1px solid black;  
}
```

```
.em_length {
    width: 20em;
    background-color: white;
    color: red;
    border: 1px solid black;
}
```

Rezultat je prikazan na slici:



z-index – Svojstvo određuje raspored elemenata u z osi. Prilikom preklapanja elemenata određuje koji element će biti prikazan preko drugog. Elementi sa većom vrednosti svojstva **z-index**, prekrivaju element sa nižom vrednosti.

Primer HTML koda:

```
<div class="dashed-box">Dashed box
    <span class="gold-box">Gold box</span>
    <span class="green-box">Green box</span>
</div>
```

Primer CSS koda:

```
.dashed-box {
    position: relative;
    z-index: 1;
    border: dashed;
    height: 8em;
    margin-bottom: 1em;
    margin-top: 2em;
}
.gold-box {
```

```

    position: absolute;
    z-index: 3; /* put .gold-box above .green-box and
.dashed-box */
    background: gold;
    width: 80%;
    left: 60px;
    top: 3em;
}
.green-box {
    position: absolute;
    z-index: 2; /* put .green-box above .dashed-box */
    background: lightgreen;
    width: 20%;
    left: 65%;
    top: -25px;
    height: 7em;
    opacity: 0.9;
}

```

Rezultat je prikazan na slici:



Medija upiti (eng. media query)

Medija upiti su korisni ako želite da modifikujete sajt ili aplikaciju u zavisnosti od generalnih ili specifičnih karakteristika i parametara uređaja, kao što su rezolucija ekrana ili širina browser-ovog prikaza (eng. viewport-a). Medija upiti se koriste za:

- Uslovnu primenu stilova sa CSS-om „@media“ i „@import“ „@-pravila“ (eng. at-rules).
- Ciljanje specifičnog sadržaja za „<link>“, „<source>“ i ostale HTML elemente.
- Testiranje i nadgledanje medija stanja koristeći Window.matchMedia() i MediaQueryList.addListener() JavaScript metode.

Napomena: primeri će koristiti CSS-ov @media za ilustrativnu svrhu, ali će osnovna sintaksa ostati ista za sve ostale tipove media upita.

Medija tipovi opisuju generalnu kategoriju uređaja. Medija tipovi su opcionalni i „all“ tip će biti primenjen, osim ukoliko se ne upotrebe „not“ ili „only“ logički operatori. Mogu se podeliti na:

- **all** – Odgovara svim uređajima.
- **print** – Namenjen je štampanom materijalu i dokumentima koji se gledaju na ekranu u „print preview“ režimu.
- **screen** – Namenjen je prvenstveno ekranima.
- **speech** – Namenjen je sintetizatorima govora.

Medija funkcije opisuju specifične funkcije browser-a, uređaja na kom se prikazuje ili okruženja. Opcione su i moraju biti okružene zagradama. Neke od medija funkcija su:

width	Širina prostora za prikaz na uređaju.
height	Visina prostora za prikaz na uređaju.
aspect-ratio	Odnos širine i visine prostora za prikaz na uređaju.
orientation	Orijentacija prostora za prikaz na uređaju.
resolution	Gustina piksela na uređaju.
update	Koliko često se može ažurirati sadržaj.
overflow-block	Kako uređaj izlazi na kraj sa sadržajem koji je veći od prostora za prikaz na uređaju.
overflow-inline	Da li višak sadržaja može da se skroluje uz osu.

Logički operatori „and“, „not“ i „only“ mogu da naprave kompleksan medija upit. Takođe možete da upotrebite više medija upita i da napravite jedan, odvajajući ih zarezima.

Operator „**and**“ služi za kombinovanje više medija opcija i pravljenje jednog medija upita i zahteva da svi delovi upita vrate vrednost „true“ da bi se upit izvršio.

Operator „**not**“ se koristi za negaciju medija upita, a ako se koristi mora da sadrži i medija tip.

Operator „**only**“ se koristi isključivo ako se ceo upit poklapa i koristan je u sprečavanju starih browser-a da primenjuju stilove, a ako se koristi mora da sadrži i medija tip.

„ , “ **zarez** se koristi prilikom kombinovanja više medija upita u jedno pravilo. Svaki upit odvojen zarezom se posmatra odvojeno od ostalih, tj. zarez se ponaša kao logički „ili“ (eng. or) operator.

Medija tipovi opisuju generalnu kategoriju datog uređaja. Iako se web sajtovi dizajniraju za prikazivanje na ekranu, data vam je opcija da dizajnirate za specijalne uređaje kao što su štampači ili čitači knjiga.

CSS koji cilja štampače je:

```
@media print { ... }
```

CSS koji cilja i ekrane i štampače je:

```
@media screen, print { ... }
```

CSS koji cilja uređaje maksimalne širine prikaza manje od 4560 piksela je:

```
@media (max-width: 4560px) { ... }
```

CSS koji koristi dve medija funkcije da ograniči upotrebu stilova na uređajima koji imaju širinu bar 30em i landscape orijentaciju je:

```
@media (min-width: 30em) and (orientation: landscape) {  
... }
```

CSS koji ima isti uslov kao prethodni primer i uz sve to i omogućuje upotrebu isključivo na uređajima sa ekranom je:

```
@media screen and (min-width: 30em) and (orientation: landscape) { ... }
```

Sledeće CSS pravilo će primeniti stilove ako uređaj ima minimalnu visinu od 680px ili ako je uređaj u portrait režimu:

```
@media (min-height: 680px), screen and (orientation: portrait) { ... }
```

„not“ ključna reč invertuje značenje celog medija upita.

Napomena: Uradiće negaciju samo specifičnog medija upita, tj. neće se primeniti na svaki medija upit u listi medija upita odvojenih zarezom. „not“ se primenjuje poslednji u upitu i sledeća dva upita imaju isto značenje:

```
@media not all and (monochrome) { ... }
```

```
@media not (all and (monochrome)) { ... }
```

Još jedan primer upita sa istim značenjem je:

```
@media not screen and (color), print and (color) { ... }  
@media (not (screen and (color))), print and (color) { ... }
```

Ključna reč „only“ sprečava starije browser-e koji ne podržavaju medija upite, da primenjuju stilove. Ovo nema efekta na nove browser-e:

```
@media (only screen and (color)) { ... }
```

CSS vrednosti i jedinice

Postoji mnogo tipova CSS svojstava: numeričke vrednosti, boje i funkcije koje vrše akcije kao što su dodavanje pozadinske slike ili rotiranje elementa. Neke od navedenih se oslanjaju na određene jedinice – da li je okvir širok 30 piksela, 30 centimetara ili 30 ems? U nastavku će biti prikazane uobičajene vrednosti kao što su dužina, boja i neke jednostavne funkcije. Takođe, postoje i jedinice koje se ređe koriste, kao što su: stepen i numerička vrednost bez jedinice mere. Biće detaljno opisane:

- Numeričke vrednosti.
- Procenti – se mogu koristiti za određivanje veličine ili dužine – relativne u odnosu na širinu i visinu okvir-a u kom se nalaze, ili za određivanje podrazumevane veličine fonta.
- Boje – Za određivanje pozadinskih boja, boje teksta i ostalo.
- Koordinate – Za određivanje pozicije elementa, relativno u odnosu na gornji levi ćošak ekrana.
- Funkcije - Za određivanje pozadinskih slika ili gradijenata pozadinskih slika.

Numeričke vrednosti će biti objašnjene kroz praktičan primer.

Primer HTML koda:

```
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>  
<p>This is a paragraph.</p>
```

Primer CSS koda:

```
p {  
    margin: 5px;  
    padding: 10px;  
    border: 2px solid black;  
    background-color: cyan;  
}  
  
p:nth-child(1) {  
    width: 150px;  
    font-size: 18px;  
}  
  
p:nth-child(2) {
```

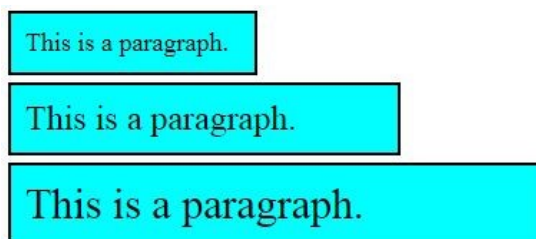
```

width: 250px;
font-size: 24px;
}

p:nth-child(3) {
width: 350px;
font-size: 30px;
}

```

Rezultat je prikazan na slici:



Navedeni kod radi sledeće:

- Podešava svojstva: „margin“, „padding“ i „border-width“ svakog paragrafa na 5 piksela, 10 piksela i 2 piksela, respektivno. Jedna vrednost za „margin“/„padding“ podešava vrednost za sve četiri strane.
- Podešava širinu tri različita paragrafa na sve veće i veće vrednosti.
- Podešava veličinu fonta tri različita paragrafa na sve veće i veće vrednosti.

Pikseli (px) su apsolutne jedinice zato što su uvek iste veličine, bez obzira na ostala podešavanja. Ostale apsolutne jedinice su:

- **mm** (milimetri),
- **cm** (centimetri),
- **in** (inči),
- **pt** (eng. points) 1/72 inča,
- **pc** (eng. picas) 12 point-a.

Najverovatnije nećete imati potrebu da koristite bilo koju drugu apsolutnu jedinicu, osim piksela.

Prisutne su i relativne jedinice, koje su zavisne od veličine fonta ili „viewport“-a:

- **em** – 1 em je isti kao i veličina fonta trenutnog elementa (preciznije, širine velikog slova M). Podrazumevana veličina fonta pre nego što se CSS stil

primeni je 16 piksela, što znači da podrazumevana izračunata vrednost 1em-a iznosi 16 piksela. em je najčešće korišćena relativna jedinica.

- **ex** – ima visinu malog slova x.
- **ch** – ima širinu broja 0.
- **rem** – root em radi na isti način kao i em, osim što će uvek primeniti podrazumevanu vrednost veličine fonta. Mnogo je bolja od opcija, ali ne radi u starim verzijama Internet Explorer-a.
- **vw** – 1/100 deo od širine viewport-a.
- **vh** – 1/100 deo od visine viewport-a.

Korišćenje relativnih jedinica je korisno zato što se HTML elementi mogu skalirati na osnovu fonta ili na osnovu vidljivog prostora (eng. viewport-a), što znači da će raspored ostati isti i ako se veličina teksta udvostruči.

Ponekad se koriste vrednosti bez jedinice mere. Npr. prilikom uklanjanja svojstva „margin“ ili „padding“ iz elementa može se zadati samo 0, bez obzira koja je jedinica ranije korišćena.

Primer HTML koda:

```
<p>Blue ocean silo royal baby space glocal evergreen  
relationship housekeeping  
native advertising diversify ideation session.  
Soup-to-nuts herding cats revolutionary  
virtuoso granularity catalyst wow factor loop  
back brainstorm. Core competency  
baked in push back silo irrational exuberance  
circle back roll-up.</p>
```

Primer CSS koda:

```
p {  
  line-height: 1.5;  
}
```

Rezultat je prikazan na slici:

Blue ocean silo royal baby space glocal evergreen relationship
housekeeping native advertising diversify ideation session.
Soup-to-nuts herding cats revolutionary virtuoso granularity
catalyst wow factor loop back brainstorm. Core competency
baked in push back silo irrational exuberance circle back roll-
up.

U prethodnom kodu je veličina fonta 16 piksela, visina linije je 1.5, što iznosi 24 piksela.

Primer animacije je dat na paragrafu, koji rotira prilikom prelaska kursora preko njega. Primer HTML koda:

```
<p>Hello</p>
```

Primer CSS koda:

```
@keyframes rotate {  
  0% {  
    transform: rotate(0deg);  
  }  
  
  100% {  
    transform: rotate(360deg);  
  }  
}  
  
p {  
  color: red;  
  width: 100px;  
  font-size: 40px;  
  transform-origin: center;  
}  
  
p:hover {  
  animation-name: rotate;  
  animation-duration: 0.6s;
```

```
    animation-timing-function: linear;
    animation-iteration-count: 5;
}
```

Deo rezultata je prikazan na slici:



Upotrebom procenata može se odrediti većina stvari koja koristi numeričke vrednosti. To omogućava da se kreiraju okviri koji će biti veličine određen broj procenata okvira u kome se nalaze.

Primer HTML koda:

```
<div>
  <div class="boxes">Fixed width layout with pixels</div>
  <div class="boxes">Liquid layout with percentages</div>
</div>
```

Primer CSS koda:

```
div .boxes {
  margin: 10px;
  font-size: 200%;
  color: white;
  height: 150px;
  border: 2px solid black;
}

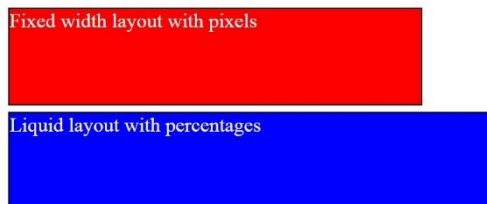
.boxes:nth-child(1) {
  background-color: red;
  width: 650px;
}

.boxes:nth-child(2) {
```



```
background-color: blue;  
width: 75%;  
}
```

Rezultat je prikazan na slici:



U ovom primeru se za oba „div“-a zadaju: margine, visina, veličina fonta, okvir i boja. Prvi dobija fiksnu širinu od 650 piksela, a drugi od 75 %. U zavisnosti od širine prikaza, mogu se značajno razlikovati širine prikaza „div“-ova, tj. „div“ koji dinamički menja širinu može na ekranu izgledati i širi i uži od prvog „div“-a. Ovo se može testirati tako što se menja širina prozora browser-a. Veličina fonta od 200 % je ekvivalentna veličini od 2 em-a. Dva prikazana rasporeda su poznatija kao tečni raspored (eng. liquid layout) i raspored fiksne širine (eng. fixed width layout). Svaki od njih ima svoju namenu:

- tečni raspored se koristi da bi bili sigurni da će dokument uvek stati na ekran i izgledati dobro na ekranima različitih veličina.
- raspored fiksne širine se koristi npr. da se mapa održi u istim dimenzijama, a da se po potrebi skroluje. Koliko se vidi odjednom, zavisi isključivo od viewport-a.

Na modernim kompjuterima je sistem boja od 24 bit-a, što omogućava prikaz 16777216 (256x256x256) boja, koja se pravi od raznih kombinacija crvene, zelene i plave (eng. RGB). Boje se mogu definisati rečima i ima ih približno 165. Za finije podešavanje moguće je koristiti:

- heksadecimalne vrednosti, npr. #0000FF,
- RGB vrednosti, npr. rgb(255,0,0) i
- HSL vrednosti, npr. hsl(240,100%,50%).
- RGBA i HSLA vrednosti, kod kojih četvrti parametar označava zamućenje (eng. opacity). A je skraćeno od „Alpha“, za vrednost 1 nema

transparentnosti, dok je za vrednost 0 element nevidljiv, primer plava boja 50%, `rgba(0,0,255,0.5)`).

Funkcije su delovi koda koji se mogu koristiti više puta za zadatke koji se ponavljaju, bez mnogo truda. Primeri funkcija su i `rgb()` i `hsl()`. Kad god vidite naziv sa zagradama posle, reč je o funkciji. Primer:

```
/* calculate the new position of an element after it
has been rotated by 45 degrees */
transform: rotate(45deg);
/* calculate the new position of an element after it
has been moved across 50px and down 60px */
transform: translate(50px, 60px);
/* calculate the computed value of 90% of the current
width minus 15px */
width: calc(90%-15px);
/* fetch an image from the network to be used as a
background image */
background-image: url('myimage.png');
```

Kombinatori

Korišćenje jednog po jednog selektora je korisno, ali u nekim situacijama može biti neefikasno. CSS ima nekoliko načina da selektuje elemente, na osnovu njihovog međusobnog odnosa. Ti odnosi se izražavaju sa kombinatorima (A i B predstavljaju selektore koje ste do sada videli):

Naziv	Sintaksa	Selektuje
Grupa selektora	A, B	Bilo koji element A ili B – ne smatra se kombinatorom
Selektor potomka (eng. Descendant selector)	A B	Bilo koji element B koji je potomak elementa A (dete, dete od deteta, itd.)
Selektor deteta (Child selector)	A > B	Bilo koji element B koji je direktni potomak elementa A
Rođeni brat/sestra selektor (eng. Adjacent sibling selector)	A + B	Bilo koji element B koji je rođeni brat/sestra elementa A (sledeće dete od istog roditelja)
Neki od rođene braće/sestara selektor (eng. General sibling selector)	A ~ B	Bilo koji element B koji je neki od rođene braće/sestara elementa A (neko od sledeće dece, od istog roditelja)

Primer HTML koda:

```
<table lang="en-US" class="with-currency">
  <thead>
    <tr>
      <th scope="col">Product</th>
      <th scope="col">Qty.</th>
      <th scope="col">Price</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th colspan="2" scope="row">Total:</th>
      <td>148.55</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Lawnchair</td>
      <td>1</td>
      <td>137.00</td>
    </tr>
    <tr>
      <td>Marshmallow rice bar</td>
      <td>2</td>
      <td>1.10</td>
    </tr>
    <tr>
      <td>Book</td>
      <td>1</td>
      <td>10.45</td>
    </tr>
  </tbody>
</table>
```

Primer CSS koda:

```
/* Basic table setup */
table {
```

```

    font: 1em sans-serif;
    border-collapse: collapse;
    border-spacing: 0;
}

/* All <td>s within a <table> and all <th>s within a
<table>
    Comma is not a combinator, it just allows you to
target
    several selectors with the same CSS ruleset */
table td, table th {
    border : 1px solid black;
    padding: 0.5em 0.5em 0.4em;
}

/* All <th>s within <thead>s that are within <table>s
*/
table thead th {
    color: white;
    background: black;
}

/* All <td>s preceded by another <td>,
    within a <tbody>, within a <table> */
table tbody td + td {
    text-align: center;
}

/* All <td>s that are a last child,
    within a <tbody>, within a <table> */
table tbody td:last-child {
    text-align: right
}

/* All <th>s, within a <tfoot>s, within a <table> */
table tfoot th {
    text-align: right;
    border-top-width: 5px;
}

```

```

border-left: none;
border-bottom: none;
}

/* All <td>s preceded by a <th>, within a <table> */
table th + td {
text-align: right;
border-top-width: 5px;
color: white;
background: black;
}

/* All pseudo-elements "before" <td>s that are a last
child,
    appearing within elements with a class of "with-
currency" that
    also have an attribute "lang" with the value "en-
US" */
.with-currency[lang="en-US"] td:last-child::before {
content: '$';
}

/* All pseudo-elements "after" <td>s that are a last
child,
    appearing within elements with the class "with-
currency" that
    also have an attribute "lang" with the value "fr"
*/
.with-currency[lang="fr"] td:last-child::after {
content: ' €';
}

```

Rezultat je prikazan na slici:

Product	Qty.	Price
Lawnchair	1	\$137.00
Marshmallow rice bar	2	\$1.10
Book	1	\$10.45
Total:		\$148.55

Pseudo selektori (pseudo klase i pseudo elementi)

Pseudo selektori ne selektuju elemente, već određene delove elemenata ili elemente u određenom kontekstu, dva glavna tipa su pseudo klase i pseudo elementi.

Pseudo klasa je ključna reč dodata na kraj selektora, odmah nakon dvotačke „:“ i određuje stilizovanje određenog elementa, u određenom stanju. Npr. hoćete da stilizujete link prilikom prelaska kursora preko njega. Neki od pseudo klasa su:

- :active ,
- :first-child,
- :hover,
- :last-child,
- :link ,
- :nth-child() ,
- :nth-of-type() i
- :visited .

Primer HTML koda:

```
<ul>
  <li><a href="#">United Kingdom</a></li>
  <li><a href="#">Germany</a></li>
  <li><a href="#">Finland</a></li>
  <li><a href="#">Russia</a></li>
  <li><a href="#">Spain</a></li>
  <li><a href="#">Poland</a></li>
</ul>
```

Primer CSS koda:

```
ul {  
    padding: 0;  
}  
  
li {  
    padding: 3px;  
    margin-bottom: 5px;  
    list-style-type: none;  
}  
  
a {  
    text-decoration: none;  
    color: black;  
}  
  
a:hover {  
    text-decoration: underline;  
    color: red;  
}  
  
li:nth-of-type(even) {  
    background-color: #ccc;  
}  
  
li:nth-of-type(odd) {  
    background-color: #eee;  
}
```

Rezultat je prikazan na slici:

United Kingdom
Germany
Finland
<u>Russia</u>
Spain
Poland

Pseudo elementi podsećaju na pseudo klase, uz razliku da pre njih slede dve dvotačke: „::“. Pseudo elementi su:

- ::after ,
- ::before ,
- ::first-letter ,
- ::first-line ,
- ::selection i
- ::backdrop .

Primer HTML koda:

```
<p>This is my very important paragraph. I am a distinguished gentleman of such renown that my paragraph needs to be styled in a manner befitting my majesty. Bow before my splendour, dear students, and go forth and learn CSS!</p>
```

Primer CSS koda:


```
p::first-line {
    font-weight: bold;
}

p::first-letter {
    font-size: 3em;
    border: 1px solid black;
    background: red;
    display: block;
```



```
float: left;  
padding: 2px;  
margin-right: 4px;  
}
```

Rezultat je prikazan na slici:



his is my very important paragraph. I am a distinguished gentleman of such renown that my paragraph needs to be styled in a manner befitting my majesty. Bow before my splendour, dear students, and go forth and learn CSS!

Napredne CSS opcije

Neke od najpopularnijih CSS biblioteka su:

- Semantic UI (<https://semantic-ui.com/>) - „razvojni framework koji olakšava kreiranje responsivnih rasporeda“.
- Foundation by ZURB (<https://foundation.zurb.com/>) - „najnapredniji responsivni front-end framework“.
- Bootstrap (<https://getbootstrap.com/>) - open source set alata koje olakšavaju razvoj HTML-a, CSS-a i JavaScript-a.
- Pure (<https://purecss.io/>) - set malih, responsivnih CSS modula koje možete koristiti u svakom web projektu.

Neki od najpopularnijih jezika formatiranja su:

- SASS (**S**yntactically **A**wesome **S**tyl**S**heets) (<http://sass-lang.com/>) – ekstenzija CSS-a koja proširuje osnovni set mogućnosti.
- LESS (**L**eaner **S**tyl**S**heets) (<http://lesscss.org/>) - unazad kompatibilna ekstenzija CSS-a.
- STYLUS (<http://stylus-lang.com/>) - inovativni jezik stilova koji se kompajlira nadole, do CSS-a.

Interesantna CSS biblioteka je „animate.css“ i moguće ju je preuzeti na :

<https://daneden.github.io/animate.css/> .

Fontovi koji će biti korišćeni u praktikumu se mogu preuzeti sa :

<https://ionicons.com/v2/> .

CSS prefiksi

Omogućavanje starim browser-ima da rade sa CSS-om, <https://www.jsdelivr.com/package/npm/respond.js> .

Kod :

```
<script  
src="https://cdn.jsdelivr.net/npm/respond.js@1.4.2/dest/respond.min.js"></script  
>
```

U zastarelim browserima je moguće obezbediti osnovne HTML stilove :

<https://www.jsdelivr.com/package/npm/html5shiv>

Kod:

```
<script  
src="https://cdn.jsdelivr.net/npm/html5shiv@3.7.3/dist/html5shiv.min.js"></script  
>
```

Moguće je emuliranje CSS 3 pseudo klasa i elemenata u starim Internet Explorer-ima: <https://www.jsdelivr.com/package/npm/selectivizr>

Kod:

```
<script  
src="https://cdn.jsdelivr.net/npm/selectivizr@1.0.3b/selectivizr.min.js"></script>
```

jQuery

jQuery je najpopularnija JavaScript biblioteka na svetu.

Kod:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Neki od interesantnih JQuery plug in-ova su Magnific Popup, One page scroll i Waypoints.

JavaScript

Osnove JavaScript-a

JavaScript je punopravni programski jezik koji nakon primene na HTML dokument može da doda interaktivnost na web sajtove. JavaScript je jako svestran. Možete započeti skromno, sa galerijama slika, dinamičkim okvirima ili odgovorima na klikove dugmića. Sa više iskustva, možete praviti igre, 2D i 3D grafiku, kompleksne aplikacije koje se oslanjaju na bazu i još mnogo toga. JavaScript je kompaktan i veoma je fleksibilan. Programeri su napisali velik broja alata u JavaScript jeziku, oslobađajući ogromnu količinu funkcionalnosti sa minimalnim ulaganjem truda, to su:

- Aplikativni programski interfejs browser-a (API) – API-ji ugrađeni u web browser-e, koji pružaju funkcionalnost kao što je dinamičko kreiranje HTML-a i postavljanje CSS stilova, skupljanje i upravljanje video strimovima sa web kamera korisnika ili generisanje 3D grafike i audio uzoraka.
- API-ji trećih strana (eng. third-party APIs) omogućavaju programerima da pripoje u njihove sajtove funkcionalnosti drugih provajdera sadržaja, kao što su Twitter ili Facebook .
- Biblioteke i eng. frameworks trećih strana, koji mogu biti primenjeni na HTML sa ciljem ubrzanog razvoja sajtova i aplikacija.

JavaScript je jedna od najdinamičnijih web tehnologija i kako budete postajali dobri u korišćenju, vaši sajtovi će ući u novu kreativnu dimenziju. Navikavanje na JavaScript je malo teže u odnosu na navikavanje na HTML i CSS. Bitno je da započnete polako i da se krećete sitnim i konzistentnim koracima. Kao početni primer, osnovni JavaScript će biti dodat u stranicu i biće kreiran „Zdravo svete. / Hello world. / Здравствуй, мир.“ primer.

Fajlovi koji će biti korišćeni su index.html, style.css i FTN.jpg iz dela: „Osnove CSS-a“, uz napomenu da se „style.css“ nalazi u folderu „styles“, a „FTN.jpg“ u folderu „images“.

Prvi korak je kreiranje foldera „scripts“ (bez navodnika) i kreiranje novog fajla „main.js“ unutar ovog foldera. Zatim u html fajlu, neposredno pre zatvarajuće „</body>“ oznake je potrebno dodati sledeći kod:

```
<script src="scripts/main.js"></script>
```

U principu, navedeni kod ima istu funkciju kao i „<link>“ element u CSS-u – on primenjuje JavaScript na stranicu i na taj način ima uticaj na HTML (pored CSS-a i svega ostalog na strani).

U „main.js“ je potrebno upisati sledeći kod:

```
var myHeading = document.querySelector('h1');
```

```
myHeading.textContent = 'Zdravo svete. / Hello world. / Здравствуй,  
мир.';
```

Navedeni kod je omogućio da se programski izmeni sadržaj „<h1>“ elementa, čiji je izgled prikazan na slici.



Napomena: razlog zbog koga je „<script>“ element skoro na kraju HTML fajla je taj što browser učitava HTML redosledom u kom se pojavljuje u fajlu. Ako se prvo učitava JavaScript koji treba da ima uticaj na određeni element a element još nije učitán, može se desiti da nešto ne radi. Zato je najbolja praksa da se JavaScript učitava pred sam kraj.

Tekst naslova je izmenjen pomoću JavaScript-a, korišćenjem funkcije „querySelector()“, koja je dohvatila referencu naslova i smestila je u promenljivu „myHeading“. Ovo je veoma slično kao što smo radili sa CSS selektorima. Ukoliko hoćemo nešto da uradimo sa elementom, moramo prvo da ga selektujemo. Nakon toga možemo vršiti izmenu, na ovom primeru to je svojstvo „textContent“ promenljive „myHeading“ koje ima novu vrednost: „Zdravo svete. / Hello world. / Здравствуй, мир.“.

Napomena: Obe korišćene funkcionalnosti su deo DOM-a (eng. Document Object Model), koji omogućava upravljanje dokumentima.

U nastavku će biti objašnjene osnovne funkcionalnosti JavaScript jezika, a vredni spomenuti da su mogućnosti o kojim će se pričati zajedničke za sve programske jezike. Ideja je da se nauče ove osnove i da se može programirati bilo šta.

Promenljive služe za čuvanje vrednosti. Ključna reč za deklarisanje promenljive je „var“, za njom sledi ime:

```
var myVariable;
```

Napomena: Tačka-zarez „;“ na kraju linije označava gde se izraz (eng. statement) završava. Apsolutno je potrebna jedino kada postoji potreba da se razdvoji više izraza u istoj liniji. Iz nekog razloga, ljudi smatraju da je dobra praksa za stavljanje tačka-zarez na kraj svake linije. Postoje pravila kada sa se koriste, a kada ne.

Napomena: JavaScript razlikuje mala i velika slova (eng. case sensitive) i tretira promenljive „myVariable“ i „myvariable“ kao dve različite. Ukoliko postoji problema u kodu, može se obratiti dodatna pažnja na promenljive.

Nakon deklarisanja promenljive, može joj se zadati vrednost:

```
myVariable = 'Ivan';
```

Moguće je i u istoj liniji deklarirati promenljivu i zadati joj vrednost:

```
var myVariable = 'Ivan';
```

Moguće je vratiti vrednost zvanjem promenljive po imenu:

```
myVariable;
```

Vrednost promenljive se može menjati i nakon zadavanja:

```
var myVariable = 'Ivan';  
myVariable = 'Ivan G.';
```

Važno je napomenuti da promenljive mogu imati vrednosti različitih tipova.

Promenljiva	Objašnjenje	Primer
String	Sekvenca teksta poznatija kao string. Da bi naznačili da je vrednost string, potrebno je da je obuhvatimo sa jednostrukim navodnicima.	<pre>var myVariable = 'Ivan';</pre>
Number	Broj. Brojevi nemaju navodnike pored.	<pre>var myVariable = 10;</pre>
Boolean	True/False vrednost. Reči true i false su specijalne ključne reči u JavaScript-u i ne zahtevaju navodnike	<pre>var myVariable = true;</pre>
Array	Struktura koja dozvoljava više vrednosti u jednoj referenci.	<pre>var myVariable = [1,'Ivan','G',10]; Pojedinačnim elementima se pristupa: myVariable[0], myVariable[1] itd.</pre>
Object	Praktično bilo šta. Sve u JavaScript-u je objekat i može biti sadržan u	<pre>var myVariable = document.querySelector("h1"); Kao i svi navedeni primeri.</pre>

	promenljivoj. Potrebno je ovo držati na umu u toku učenja.	
--	--	--

Promenljive su potrebne da bi se uradilo bilo šta interesantno u programiranju. Kada se vrednosti ne bi mogle menjati, ne bi moglo da se radi ništa dinamično, kao što je personalizovanje pozdravne poruke ili menjanje slike u galeriji slika.

Komentar u JavaScript-u se može napisati kao i u CSS-u:

```
/*
Sve između je komentar
*/
```

Ukoliko komentar može stati u jednu liniju može se napisati kao:

```
// Ovo je komentar
```

Operator je matematički simbol koji proizvodi rezultat zasnovan na dve vrednosti (ili promenljive). U tabeli se nalazi spisak najjednostavnijih operatora, uz primere.

Operator	Objašnjenje	Simbol(i)	Primer
Sabiranje	Koristi se za sabiranje brojeva i za spajanje string-ova	+	6 + 9; "Hello " + "world!";
Oduzimanje, množenje i deljenje	Matematičke operacije	-, *, /	9 - 3; 8 * 2; // U JavaScript-u je * znak za množenje 9 / 3;
Dodela	Dodeljuje vrednost promenljivoj.	=	var myVariable = 'Ivan';
Jednakost	Testira da li su dve vrednosti jednake i vraća true/false (Boolean) rezultat.	===	var myVariable = 3; myVariable === 4;
Negacija, Nije jednako	Vraća logički suprotnu vrednost od onoga ispred čega se nalazi. Pretvara true u false, itd. Kada se stavi ispred operatora jednakosti, operator negacije testira da li dve	!, !==	Osnovni izraz je true, ali rezultat poređenja vraća false, zato što smo primenili negaciju. var myVariable = 3; !(myVariable === 3); Ovde se testira da li promenljiva nije jednaka

	vrednosti nisu jednake.		broju 3. Rezultat je false, zato što promenljiva je jednaka broju tri. var myVariable = 3; myVariable !== 3; Rezultat je false.
--	-------------------------	--	--

Postoji još mnogo operatora.

Napomena: Mešanje tipova može da proizvede neobične rezultate prilikom računanja i zato je potrebno biti pažljiv prilikom pozivanja promenljivih, da bi dobili željene rezultate. Npr. ukoliko se upiše "35" + "25", kao rezultat će se dobiti "3525", iz razloga što upotrebom navodnika se brojevi pretvaraju u stringove. Ukoliko se upiše 35 + 25, vrši se operacija sabiranja.

Uslovi su strukture koda koji omogućavaju da se izvrši planirani ili alternativni deo koda. Najčešća forma uslova je if...else izraz, npr:

```
var iceCream = 'chocolate';
if (iceCream === 'chocolate') {
    alert("Yay, I love chocolate ice cream!");
} else {
    alert('Awww, but chocolate is my favorite...');
}
```

Izraz unutar „if (...)“ je test koji koristi operator jednakosti da poredi promenljivu „iceCream“ sa string-om „chocolate“. Ako je rezultat poređenja jednak „true“, prvi blok koda će se izvršiti (unutar vitičastih zagrada). Ako rezultat poređenja nije „true“, prvi blok se preskače i ide se na drugi, posle „else“ izraza (unutar vitičastih zagrada).

Funkcije su način pakovanja funkcionalnosti, koju želite ponovo da iskoristite. Do sada su navedena dva primera funkcija:

```
var myVariable = document.querySelector("h1");

alert('hello!');
```

Funkcije „document.querySelector“ i „alert“ su ugrađene u browser-e. Ako vidite nešto što liči na promenljivu i ima zagrade () na kraju, najverovatnije je reč o funkciji. Funkcije često uzimaju argumente, kao parčiće podataka koje su im potrebne da nešto izvrše. Argumenti se pišu unutar zagrada, a ako ih je više razdvajaju se zarezom. Npr. „alert()“ funkcija omogućava da se iskačući prozor pojavi u browser-u, a potrebno je definisati šta će biti prikazano. Dobra vest je što

je moguće definisati sopstvenu funkciju koja uzima dva broja kao argumente i množi ih:

```
function multiply(num1,num2) {  
    var result = num1 * num2;  
    return result;  
}
```

Navedena funkcija se poziva na sledeći način:

```
multiply(4,7);
```

Napomena: „return“ izjava govori browser-u da vrati result promenljivu iz funkcije i da je učini dostupnom. Ovo je potrebno zato što su promenljive definisane u funkcije jedino dostupne unutar tih funkcija. Ovo se zove opseg promenljivih.

Prava interaktivnost na web sajtu se postiže sa događajima. To su strukture koda koje slušaju šta se dešava u browser-u i kao odgovor izvršavaju određeni kod. Najočigledniji primer je klik događaj (eng. click event), koji browser pokreće kada se mišem klikne na nešto:

```
document.querySelector("html").onclick = function() {  
    alert('Click.');
```

```
}
```

Postoji više načina da se događaj (eng. event) zakači za element. Ovde selektujemo „<html>“ element i podešavamo da je svojstvo „onclick“ handler-a anonimna funkcija (funkcija bez imena), koja sadrži kod kog želimo da pokrenemo.

Može se primetiti da je kod:

```
document.querySelector("html").onclick = function() {};
```

ekvivalentan sa:

```
var myHTML = document.querySelector("html");  
myHTML.onclick = function() {};
```

Kod napisan na prvi način je kraći.

U sledećem primeru će biti dodata još jedna slika koja će uz pomoć DOM API funkcionalnosti i korišćenja JavaScript-a moći da bude izmenjena, prilikom klika na sliku. Potrebno je kreirati novu sliku, približnih dimenzija i snimiti je u folder „images“ i označiti je kao „FTN2.jpg“. U „main.js“ potrebno je zameniti postojeći kod sa:

```

var myImage = document.querySelector('img');

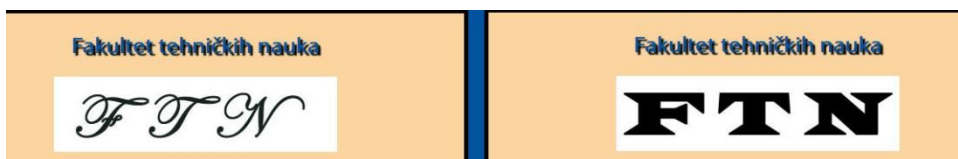
myImage.onclick = function() {
    var mySrc = myImage.getAttribute('src');
    if(mySrc === 'images/FTN.jpg') {
        myImage.setAttribute ('src','images/FTN2.jpg');
    } else {
        myImage.setAttribute ('src','images/FTN.jpg');
    }
}

```

Nakon snimanja izmena i pokretanja „index.html“ u browser-u, vidi se da prilikom klika na sliku pokazuje ona druga. Referenca „“ elementa se čuva u „myImage“ promenljivoj. Svojstvu promenljive, „onclick“ event handler-u se zadaje anonimna funkcija. Nakon klika na element vrši se:

- Vraćanje vrednosti „src“ atributa od slike.
- Provera preko uslova da li je vrednost „src“ atributa jednaka putanji originalne slike:
 - Ako jeste, vrednost „src“ elementa se menja na putanju prema drugoj slici, čime se omogućava učitavanje slike u „“ element.
 - Ako nije (znači da se već promenila), „src“ vrednost se vraća na originalnu putanju.

Rezultati ubačenog JavaScript-a koji omogućava promenu slike klikom na istu, su prikazani na slici:



Sledeći primer će zameniti naslov stranice personalizovanom pozdravnom porukom, prilikom prve posete stranici. Ova pozdravna poruka će ostati i ako korisnik ode sa stranice i kasnije se vrati – biće snimljena uz pomoć: „Web Storage API“. Biće omogućeno da se promeni ime korisnika, a i pozdravna poruka, kad god to bude zahtevano.

U „index.html“ neposredno pre „<a>“ elementa potrebno je dodati sledeći kod:

```

<button>Drugi korisnik</button>

```

U „main.js“ na kraju fajla potrebno je dodati sledeći kod:

```
var myButton = document.querySelector('button');  
var myHeading = document.querySelector('h1');
```

Sledeću funkciju je potrebno dodati da bi se omogućila personalizovana pozdravna poruka, kod je :

```
function setUsername() {  
    var myName = prompt('Upišite svoje ime:');  
    localStorage.setItem('name', myName);  
    myHeading.textContent = 'Ulogovani korisnik je: ' + myName;  
}
```

Ova funkcija sadrži „prompt()“ funkciju koja omogućava prozor sa dijalogom (eng. dialog box), nešto poput „alert()“. Funkcija „prompt()“ ipak pita korisnika da upiše određene podatke, koji će biti sačuvani u promenljivu nakon što korisnik pritisne „OK“. U ovom slučaju tražimo od korisnika da upiše ime. Poziva se API „localStorage“, koji omogućava da se podaci sačuvaju u browser-u i da im kasnije možemo pristupiti. Funkcija „setItem()“ kreira i čuva podatak pod nazivom „name“ i postavlja mu promenljivu „myName“ kao vrednost. Promenljiva „myName“ sadrži podatke koje je korisnik upisao. Nova vrednost svojstva „textContent“ se sastoji od string-a i imena koje je korisnik upisao. Sledeći korak je dodavanje „if...else“ bloka. Možemo ga nazvati kodom za inicijalizaciju, pošto strukturu aplikaciju prilikom prvog učitavanja, kod je:

```
if(!localStorage.getItem('name')) {  
    setUsername();  
} else {  
    var storedName = localStorage.getItem('name');  
    myHeading.textContent = 'Ulogovani korisnik je: ' + storedName;  
}
```

Ovaj kod koristi operator negacije (logičko ne, koje je predstavljeno sa „!“) da proverí da li podatak „name“ postoji. Ako ne postoji, funkcija „setUserName()“ se poziva i podatak „name“ se kreira. Ako postoji sačuvano ime (čak i iz prethodne posete), ono se vraća koristeći „getItem()“. Vrednost naslova se postavlja kao string „textContent“ sa imenom korisnika, što je urađeno unutar „setUserName()“.

Na kraju se na dugme dodaje „onclick“ event handler:

```
myButton.onclick = function() {
```

```
    setUsername();  
}
```

Pritiskom na dugme se korisniku omogućava da u bilo kom momentu promeni ime koje je prikazano u naslovu, uz pomoć funkcije „setUserName()“.

Prilikom prve posete stranici potrebno je upisati korisničko ime, a zatim će se prikazati personalizovana poruka. Ime se može promeniti u bilo kom trenutku, klikom na dugme. Bonus je to što se i nakon isključivanja browser-a prikazuje upisano ime, a razlog je što se promenljiva čuva unutar „localStorage“.

Sadržaj „main.js“ je:

```
var myImage = document.querySelector('img');  
  
myImage.onclick = function() {  
    var mySrc = myImage.getAttribute('src');  
    if(mySrc === 'images/FTN.jpg') {  
        myImage.setAttribute ('src','images/FTN2.jpg');  
    } else {  
        myImage.setAttribute ('src','images/FTN.jpg');  
    }  
}  
  
var myButton = document.querySelector('button');  
var myHeading = document.querySelector('h1');  
  
function setUsername() {  
    var myName = prompt('Upišite svoje ime:');  
    localStorage.setItem('name', myName);  
    myHeading.textContent = 'Ulogovani korisnik je: ' + myName;  
}  
  
if(!localStorage.getItem('name')) {  
    setUsername();  
} else {  
    var storedName = localStorage.getItem('name');  
    myHeading.textContent = 'Ulogovani korisnik je: ' + storedName;  
}  
  
myButton.onclick = function() {
```

```
setUserName();  
}
```

Rezultat je prikazan na slici:

